

University of Groningen

Design of a Methodology to Support Software Release Decisions

Sassenburg, J.A.

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version

Publisher's PDF, also known as Version of record

Publication date:

2006

[Link to publication in University of Groningen/UMCG research database](#)

Citation for published version (APA):

Sassenburg, J. A. (2006). *Design of a Methodology to Support Software Release Decisions: Do the Numbers Really Matter?* [Thesis fully internal (DIV), University of Groningen]. s.n.

Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

7 RELEASE INFORMATION

“When action grows unprofitable, gather information; when information grows unprofitable, sleep.”

-- Ursula K. Le Guin (1969) --

7.1 Introduction

In the previous Chapter, an economic market entry model is defined, for comparing software release alternatives. The perspective of maximizing behaviour is assumed, stemming from the theory of the firm. The primary objective of a firm, or, in this case, a software manufacturer, is to maximize long-term expected value (Hirschey 2003, p.5). For the decision-maker, it is assumed that:

- ❖ The decision-maker is an economic person with the objective of maximizing expected utility.
- ❖ The decision-maker can choose from a set of well-defined, and mutually exclusive, alternatives.
- ❖ The decision-maker is able to estimate the outcome, and calculate the expected value of each alternative. (Harrison 1987, p.79)

These assumptions stem from the closed decision model of economic theory, in its turn assuming:

- ❖ Availability of most, if not all, information relevant to the objective.
- ❖ Ability to quantify the information gathered.
- ❖ Desirability of ignoring information that is not readily quantifiable. (Harrison 1987, p.84)

In the exploratory case studies however, it was found that, in a practical setting, time and cost constraints are present, the environment is not closed but open, and, as a result, the available information for decision-making is neither complete nor reliable (Section 4.4.3). In this Chapter the effect of taking information costs into account is discussed.

In Section 7.2 the ‘economics’ of information is discussed, showing that information perfection has a price. In Section 7.3 the economics of information are applied to software release decisions, identifying the main sources of information decision-makers should seek, with an answer given to the 2nd Secondary Research Question: ‘To what extent can an optimal level of information be determined as input to the software release decision-making process?’ Based on the clarification of aspects of optimizing behaviour, practices for the ‘*Release Information*’ process area are derived in Section 7.4. This Chapter ends with a summary and conclusions in Section 7.5.

7.2 Economics of Information

Maximizing behaviour assumes that decision-makers have complete information about costs and benefits associated with each option. They compare the options on a single scale of preference, value or utility. Modern behavioural economics acknowledge however, that the assumption of perfect [complete and reliable] information is implausible. Etzioni and Amitai (1989) argue that because, normally, limitations on information will exist, it is impossible to undertake the precise analysis necessary to maximize economic objectives. Many economists put similar, and other arguments, against the case for maximizing behaviour (for example,

McQuire 1947; Simon 1955, 1957; March and Simon 1958; Winter 1964; Anthony 1966; Jensen and Meckling 1976). Rather than assuming decision-makers possess all relevant information for making choices, information is, itself, treated as a commodity, something that has a price in time and/or money.

This argument of limitations on information can be used to 'soften' maximizing behaviour to *optimizing behaviour*, where an individual decision-maker makes a trade-off between information perfection [completeness and reliability] and the cost related to searching for additional information. This relationship is described by Harrison (1987, p.49), and is given in Figure 7-1. On the horizontal axis, *Information perfection* is measured, which is knowledge about the decision outcome of an alternative. When information perfection equals 100%, the information is complete and reliable, or, supposedly, perfect. The vertical axis measures the value, cost and yield [marginal value] as a function of information perfection on the horizontal axis. *Value* refers to how desirable a particular decision outcome is considering the value of the alternative,⁶⁷ whether in money, satisfaction or other benefit. The value curve $V(i)$ rises steadily. *Cost* is the cost involved in searching for alternatives, for example, extending information perfection. The cost curve $C(i)$ moves in the opposite direction, rising rather slowly at the start because the initial information requires relatively little effort. Additional information becomes more difficult to obtain and the associated cost increases exponentially. *Yield* is the difference between value and cost [net value]. The yield curve $Y(i)$, the difference between the value and cost functions, reduces sooner, and more steeply than the value curve. *Yield* represents the net value with the point of diminishing returns, or *point of optimality* Y^* , the point where this curve reaches its maximum with the corresponding values I^* , V^* and C^* .⁶⁸ Beyond this point, the cost of acquiring additional information outweighs the value or benefit.

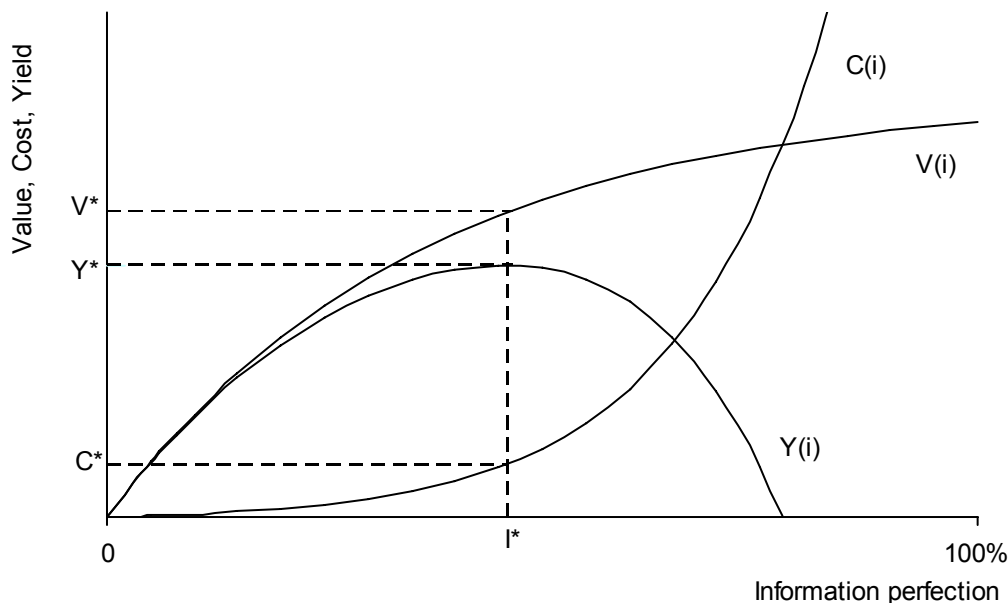


Figure 7-1: Value (V), Cost (C) and Yield (Y) as a Function of Information Perfection
(Harrison 1987, p.49)

This model can be formulated as follows:

⁶⁷ Alternatives are the possibilities one has to choose from. Alternatives can be identified [searched for and located] or even developed [created where they did not previously exist].

⁶⁸ Note that the value function $V(I)$ and cost function $C(I)$ can be expressed in different units, $V(I)$, for example, on an ordinal scale and $C(I)$ on a monetary scale. This may make it difficult to find a common denominator and thus 'calculate' the resultant yield function $Y(I)$.

1. The information perfection i ranges from:

$$0\% \leq i \leq 100\%$$

2. It is assumed the cost function $C(i)$ can be described as exponentially increasing and thus a strictly convex function with the conditions:

$$\begin{aligned} C(0) &= 0 \\ C(\infty) &= \infty \\ dC(0) / di &= 0 \\ dC(i) / di &= \text{constant} \cdot C(i) > 0 \\ d^2C(i) / di^2 &> 0 \end{aligned} \quad ^{69}$$

Trying to perfect information leads to an exponential rise in cost [diminishing marginal productivity].

3. It is assumed the value function $V(i)$ can be described as a strictly concave function with the conditions:

$$\begin{aligned} V(i) &= 0 \\ V(\infty) &= \text{constant} \\ dV(i) / di &> 0 \\ d^2V(i) / di^2 &< 0 \end{aligned}$$

The possible effect of information overflow, and its effect on the value function, is disregarded here. Information overflow occurs when marginal decreases in value, due to additional information quantity, are greater than the marginal increases in value due to additional information quality (Keller and Staelin 1987; Buchanan 2000).

4. The yield curve $Y(i)$ is given by:

$$Y(i) = V(i) - C(i) \quad (7.1)$$

The point of optimality is the point Y^* on the yield curve, with I^* being the corresponding information perfection, C^* the corresponding cost level and V^* the value of information obtained. It is the point on the yield curve $Y(i)$ where:

$$\begin{aligned} dY(i) / di &= 0 \\ &= d(V(i) - C(i)) / di \end{aligned} \quad (7.2)$$

This is the intersection of the derivatives of value and cost functions with:

$$dV(i) / dC(i) = 1 \quad (7.3)$$

Expressing V as a function of C , with the elimination of i , and calculating the derivative, the intersection of the derivatives of value and cost function can graphically confirm this, as shown in Figure 7-2.

⁶⁹ Exponential function: the slope of the cost function at any value is proportional to the value of the cost function itself.

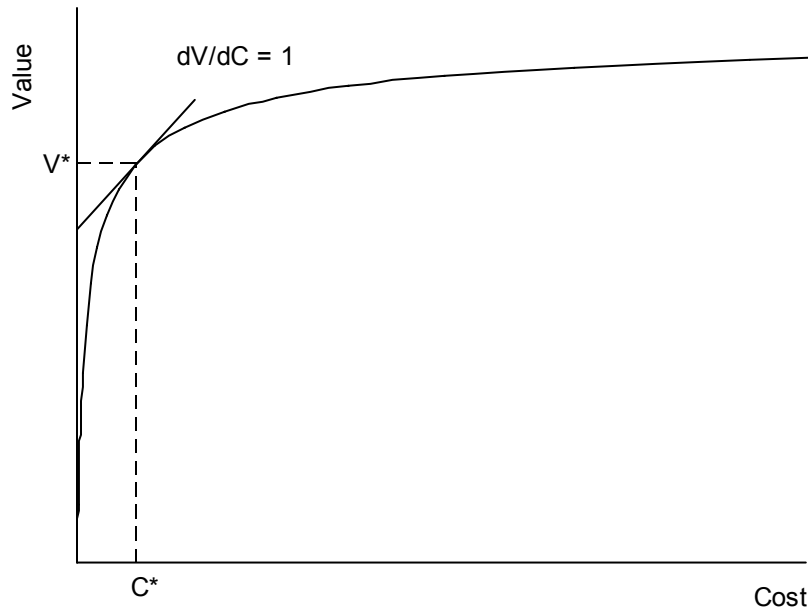


Figure 7-2: Point of Optimality

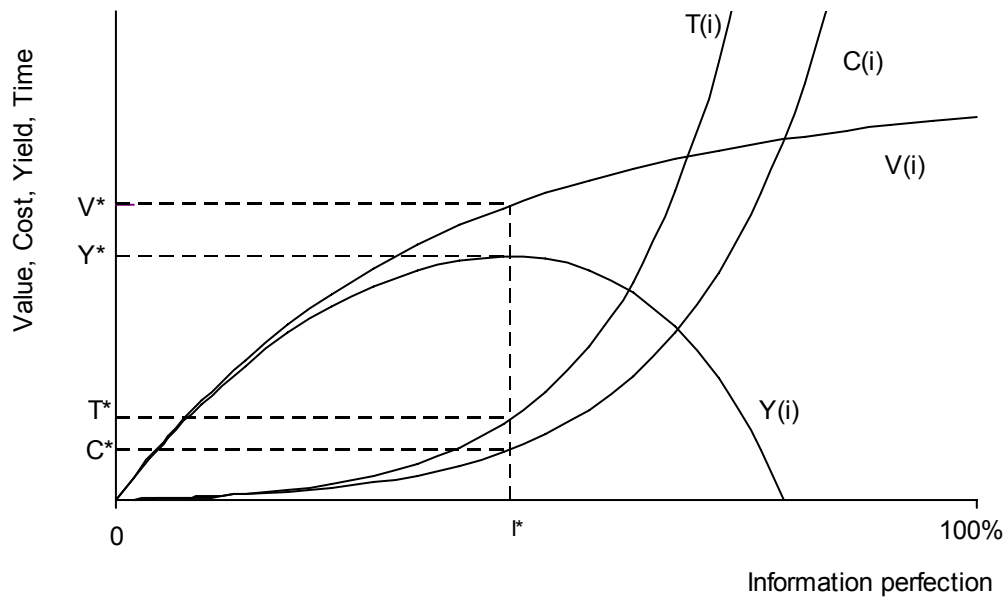


Figure 7-3: Extended Model with Time Function $[T(i)]$

Time is not taken into account. As time is also a constraint when perfecting information, the model is extended to this point by incorporating the time function $T(i)$. It is assumed this function is merely identical to the cost function $C(i)$ and can also be described as exponentially increasing and thus a strictly convex function with the conditions:

$$\begin{aligned}
 T(0) &= 0 \\
 T(\infty) &= \infty \\
 dT(0) / di &= 0 \\
 dT(i) / di &= \text{constant} \cdot T(i) > 0 \\
 d^2T(i) / di^2 &> 0
 \end{aligned}$$

Trying to perfect information will lead to an exponential rise in time, with an extended model given in Figure 7-3.^{70,71}

Prior to a further discussion on the concept of optimizing behaviour, the effects of changing the value, cost and time functions are presented. In Figure 7-4 three different value functions are given, $V_n(i)$ being the nominal function, $V_l(i)$ being the value function for a less efficient and effective information collection and exchange process, and $V_h(i)$ is the value function for a more efficient and effective information collection and exchange process.

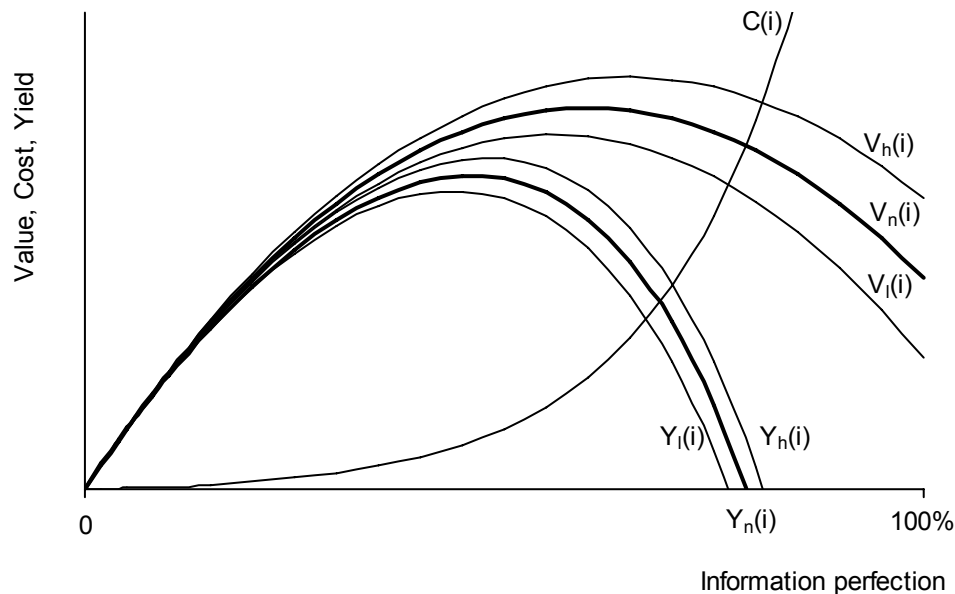


Figure 7-4: Different Value Functions

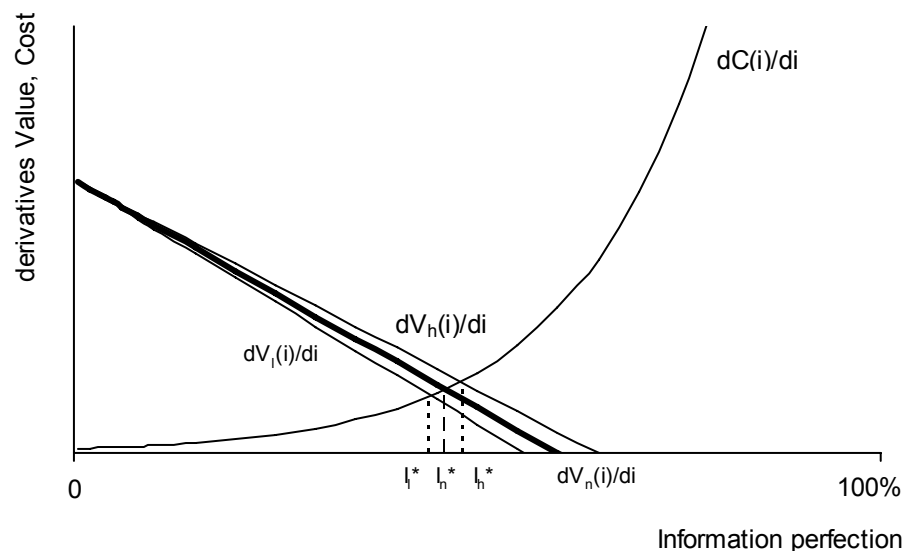


Figure 7-5: Point of Optimality

⁷⁰ Note: principally, it is possible to obtain a direct relationship between cost and time: $C = f(t)$ by taking their functions and eliminating i . In reality however, one has, to some extent, the flexibility to increase the cost level without significantly changing the time frame, for example, by letting resources work in parallel.

⁷¹ In this figure, the time function $T(i)$ rises more steeply than the cost function $C(i)$. It is however only an example as the functions, other than their basic shapes, are not known.

The resulting points of optimality for different value functions are given in Figure 7-5, using the intersection between the derivatives of the value functions and the cost function. It is clear that the point of optimality shifts to the right when the value function increases; more valuable information is obtained against equal cost.

The same exercise can be performed for different cost or time functions. The resultant points of optimality for different cost functions are given in Figure 7-6, using the intersection between the derivatives of the value function and the cost functions [$C_n(i)$ is the nominal function, $C_l(i)$ the cost function for a more efficient and effective information collection and exchange process, and $C_h(i)$ the cost function for a less efficient and effective information collection and exchange process]. The point of optimality shifts to the right when the cost function decreases, and valuable information is obtained against lower cost. Further, when the cost function decreases, the time function will probably decrease as well, as information is obtained in less time. The same effect might occur when the time function decreases, and valuable information is obtained in less time and probably against a lesser cost.

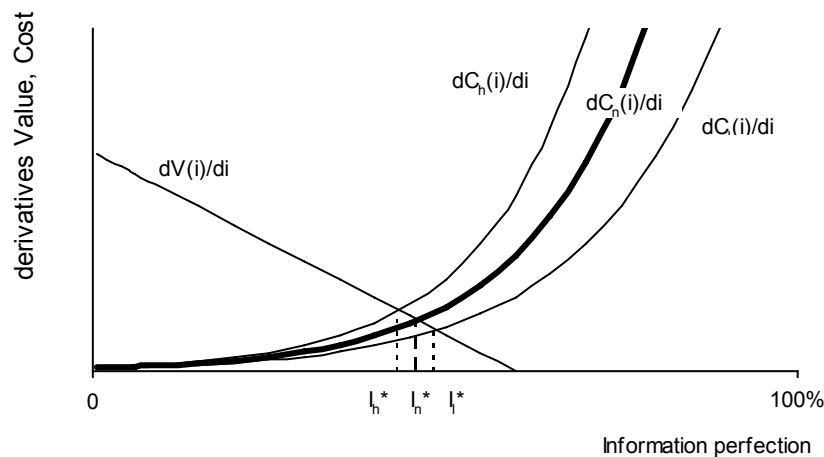


Figure 7-6: Point of Optimality ~ Different Cost Functions

These exercises are important, as improved decision-making requires the availability of additional valuable information. When more valuable information can be obtained against less cost and time, it should have the effect of shifting the point of optimality to the right, thus reducing the level of uncertainty involved in the decision. Using the definitions discussed in Section 3.4.2, improving information perfection should, theoretically, enable a decision-maker to transform a decision with complete uncertainty [point of optimality to the left] to a decision with informed uncertainty [point of optimality moving to the right] or even a decision with certainty [point of optimality completely to the right].

The model presented partially eliminates criticism on the maximizing behaviour theory: collecting information has a price in time and money, and, in a practical context, absolute perfection will normally not be sought. This leads to uncertainty. Witteloostuijn (1988, p.298) distinguishes: uncertainty of the composition of the choice set of alternative courses of action [action uncertainty], the cost and benefit of the alternatives concerned [yield uncertainty] and/or the availability of additional information and the cost and benefit of processing and gathering it [information uncertainty]. However, this concept of optimizing behaviour still assumes that the expected utility is maximized as decision-makers are, implicitly assumed, to search for the point of optimality, above which the net yield declines [law of diminishing returns]. The point of optimality is the point where marginal value equals marginal costs and where marginal yield is zero.

The difference between maximizing and optimizing behaviour is that the search for information is taken into account, as an economic activity in the classical production function, with associated costs and time. The primary objective however stays the same – profit maximization.

A criticism is that a decision-maker is now not only confronted with determining the shapes of supply and demand curves, but also the costs and benefits of searching for information. This may even magnify, and multiply, the computational complexities a decision-maker faces (Simon 1978, p.359).

Another area of concern is determining the point of optimality (Gigerenzer 2004):

- ❖ Optimization is impossible in many natural situations. In most natural situations optimization is computationally intractable, as a combinatorial explosion is generated. In such cases, the reality should be simplified (Tsotsos 1991).
- ❖ Optimization is impossible when problems are unfamiliar and time is scarce. Even when optimization is, in principle, possible, a practical problem remains. In a familiar problem, the decision-maker knows how to reach optimization [routine, simple problem], but for an unfamiliar problem there is no known method of arriving at the optimum (Selten 2001). Note that strategic software release decisions are, typically, unfamiliar problems (non-routine decisions, see Section 4.4.3). Not knowing how to find the optimum is, in fact, the problem of *infinite regress*; how can one know the expected value and cost of more information? (Elster 1983). To answer this question one needs to collect information to determine how much information to collect before one decides how much information one should collect *ad infinitum*.
- ❖ Optimization does not imply an optimal outcome. To optimize one has to make assumptions about reality, often leading to simplifications. If they are wrong or too simple, optimization will not necessarily lead to an optimal outcome (March 1978).
- ❖ A good fit, *per se*, is not an empirical validation of the model. It is not the descriptive accuracy that is important, but the accuracy of the predictions that are made (Friedman 1953).⁷²

Instead of the point of optimality a *zone of cost effectiveness* may be defined; a bandwidth where the marginal yield is equal, or close, to zero. This zone of cost effectiveness defines the relationship between monetary cost and effectiveness measured in terms of actual results. A solution is considered cost effective compared to other alternatives, if it is:

- (1) Less costly and, at least, as effective;
- (2) More costly and more effective, with an added efficacy justifying the additional price;
- (3) Less effective and less costly, where the additional cost of the alternative(s) is too high for the additional benefits provided.

In the next Section, the concept of optimizing behaviour is applied to strategic software release decisions. It is demonstrated that accepting the concept of optimizing behaviour only marginally influences the maximizing behaviour model presented in the previous Chapter.

7.3 *Applicability to Strategic Software Release Decisions*

When nearing the planned release date, the question asked by a software manufacturer is: when can testing be stopped so the product can be released? In the previous Chapter the NPVI-method was introduced to enable the comparison of different release alternatives. This method is based on comparing the net present values of different release alternatives. In this Section,

⁷² This is also one of the problems of software reliability estimation models as discussed in Section 3.3.2 (Fenton and Pfleeger 1997).

the formula used for calculating the net present value is further refined, which enables the determination of different sources of information as input to the release decision-making process.

7.3.1 Refined Model

In Section 6.4, a general formula for calculating the net present value is introduced (equation 6.16). In this Section, this formula is further refined, using the general form of software reliability estimation models, as presented in Section 3.3.2 (equation 3.11). The two higher-order limitations on these software reliability estimation models are also addressed (Section 3.3.3):

1. Focus of the models is on cash outflows, with cash inflows not taken into account. The time value of money is also discarded.
2. Focus of the models is on pre-release testing versus post-release corrective cash flows, not total cash outflows, including the additional [avoidable] costs for future product enhancements incurred for a low level of maintainability.

The limitation that only cash outflows are considered can easily be overcome by incorporating cash inflows, while the time value of money can also easily be incorporated. This results in the net present value function $NPV(t)$, as presented in Section 6.4, as the difference between the asset value [cash inflows] and two cost functions [cash outflows]. In formula form (see also equation 6.16):

$$NPV(t) = -I(t) + [C(t) - M(t)] / (1 + r)^{Tr} \quad (7.4)$$

For the pre-release operational cost function $I(t)$, the equation introduced in Section 3.3.2 is used, extended by development cost spent before testing:

$$\begin{aligned} I(t) &= I_d(T_d) + I_t(t - T_d) \\ &= I_d(t) + i_1 \cdot m(t - T_d) + i_2 \cdot (t - T_d) \end{aligned} \quad (7.5)$$

with

$I_d(T_d)$:	<i>total development cost before testing</i> ⁷³
$I_t(t)$:	<i>total cost of testing and removing faults during testing phase</i>
i_1 :	<i>expected cost of removing a fault during testing phase</i>
i_2 :	<i>expected cost per unit time of testing</i>
$m(t)$:	<i>expected mean number of faults detected in time $(T_d, t]$</i>
T_d :	<i>start of testing phase</i>

Where the test effort is constant over time [sum of effort spent on testing and fault removal is constant], the pre-release testing cost function $I_t(t)$ can be described as a linear curve and thus as an increasing, strictly convex, function. Assuming the additional cost for finding, and removing, subsequent defects increases as testing progresses, trying to perfect the product will lead to an exponential increase in development cost [diminishing marginal productivity].

For the post-release operational cost function $M(t)$, the equation $M_s(t)$, as introduced in Section 3.3.2, is used, extended with the function $M_i(t)$ illustrating the avoidable cost for future product enhancements due to a low level of maintainability. This eliminates the limitation that only short-term corrective cash outflows are considered. The exact shape of the function $M_i(t)$ is determined by factors, such as the quality of the product design [extent to which maintainability requirements are addressed], the quality of the product realization [extent to which maintainability requirements are correctly implemented], and the quality of the documentation

⁷³ Note that when comparing different release alternatives, the development costs before testing can be discarded, as they are sunk costs [non-retrievable]. It is further assumed that during testing no further development takes place.

[extent to which the product is documented in an accessible way]. The function $M(t)$ can be described as:

$$\begin{aligned} M(t) &= M_s(t) + M_l(t) \\ &= m_l \cdot [m(\infty) - m(T_r)] + M_l(t) \end{aligned} \quad (7.6)$$

with

M :	<i>total cost of testing and removing faults during operation phase and total avoidable cost for future product enhancements</i>
$M_s(t)$:	<i>total cost of testing and removing faults during operation phase (corrective maintenance)</i>
$M_l(t)$:	<i>total avoidable cost for future product enhancements due to low maintainability</i> ⁷⁴
m_l :	<i>expected cost of removing a fault during operation phase</i>
$m(t)$:	<i>expected mean number of faults detected in time $(T_r, t]$</i> ⁷⁵
T_r :	<i>release time</i>

Assuming the test effort during the testing phase is constant, and that the number of faults found per unit time of testing will decrease as testing proceeds ['easy' defects are found in the early test phase], this cost function $M(t)$ can be described as exponentially decreasing, and thus a, strictly convex, function. Trying to perfect a product leads to an exponential decrease in operational cost.

The exact shape of the newly-introduced asset value function $C(t)$ depends on the market window and the extent to which functional and non-functional requirements implemented meet customer/end-user demands. It is assumed that this asset value function $C(t)$ can be described as a strictly concave, quadratic function [first-increasing-then-decreasing] with a global maximum of C_m , conditions being:

$$\begin{aligned} C(0) &= C_a \\ dC(t)/dt &> 0 & \text{if } C < C_m \\ dC(t)/dt &= 0 & \text{if } C = C_m \\ dC(t)/dt &< 0 & \text{if } C > C_m \\ d^2C(t)/dt^2 &< 0 \\ d^3C(t)/dt^3 &= 0 \end{aligned}$$

The point of maximal net present value on the curve is the point NPV^* where its marginal value is zero, with T^* being the optimal release time, I^* the associated development cost level, M^* the associated operational cost level and C^* the asset value obtained, as graphically depicted in Figure 7-7:

$$\begin{aligned} dNPV(t)/dt &= 0 \\ &= d(-I(t) + [C(t) - M(t)] / (1 + r)^{T_r}) / dt \end{aligned} \quad (7.7)$$

⁷⁴ In Section 4.4.2, SAAM is briefly mentioned as a method to assess the maintainability of a software architecture. The *Maintainability Index*, as discussed in Section 3.3.3, or an equivalent measure, might be considered to determine the long-term maintainability of code implemented.

⁷⁵ The same mean value function $m(t)$ is used here as in equation 7.5. However, as the mean value function now concerns the operational phase, it should reflect the operational profile.

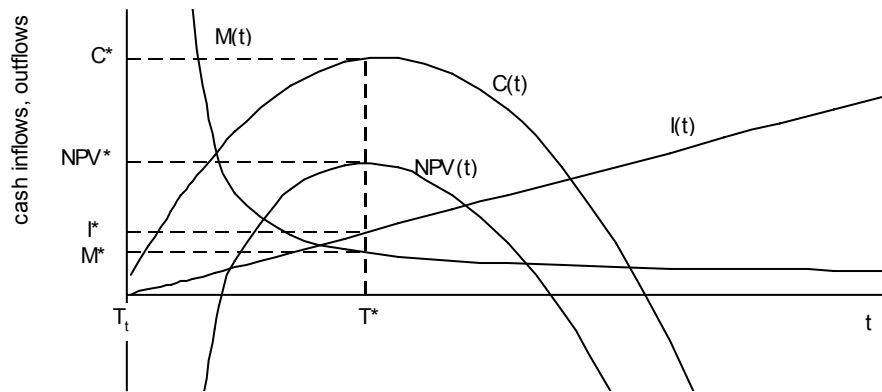


Figure 7-7: Asset Value

The exact shape of the cost function $I(t)$ depends on many factors; among which the characteristics of the *product* being developed, the development *process* being used, the underlying *technology* used [both supporting development technology and technology used in the product itself], and the capabilities and experience level of the *people* in the project. The same product developed by two different project teams, using the same process and technology, might reveal different maximal net asset values. Similarly, the same product developed by two identical project teams, using the same technology but a different process, might reveal different maximal net asset values. In an analogous way, the factors mentioned influence the shape of the post-release operational cost function $M(t)$. The same product maintained by two identical maintenance teams, using the same technology but a different maintenance process might reveal different maximal net asset values.

The model presented gives a more detailed description for determining the net present value than used in the previous Chapter. It eliminates the limitations found in available software reliability estimation models. In the next Section, this model is used to determine the sources of information as input to the software release decision-making process.

7.3.2 Optimal Information Level

Using the model from the previous Section, the main sources of information, as in Figure 7-8, as input to the decision-making process, can be derived:

1. External market-related information about demand functions [market window and expected revenues or cash inflows] is needed to estimate the shape of the asset value function $C(t)$.
2. For product-related information, two different sources are distinguished:
 - Information on the quality of the product must be obtained through verification activities. These activities should supply information on the quality of the product [implemented functional and non-functional requirements], and information on the effort needed to further improve the quality of the product. This information determines the shape of the development cost function $I(t)$; during the testing phase, and also in the development phase, prior to the testing phase.
 - Information on the maintainability of the product is needed to estimate the shape of the post-release maintenance function $M(t)$. This information should cover both the expected short-term corrective maintenance cost and the expected long-term adaptive/perfective maintenance cost.

3. Finally, information on the relationship between external market-related and internal product-related information is determined by how the quality of the product [related to $I(t)$], and the maintainability of the product [related to $M(t)$], affect the asset value function $C(t)$. In other words, how do changes in product quality and product maintainability affect the demand function?

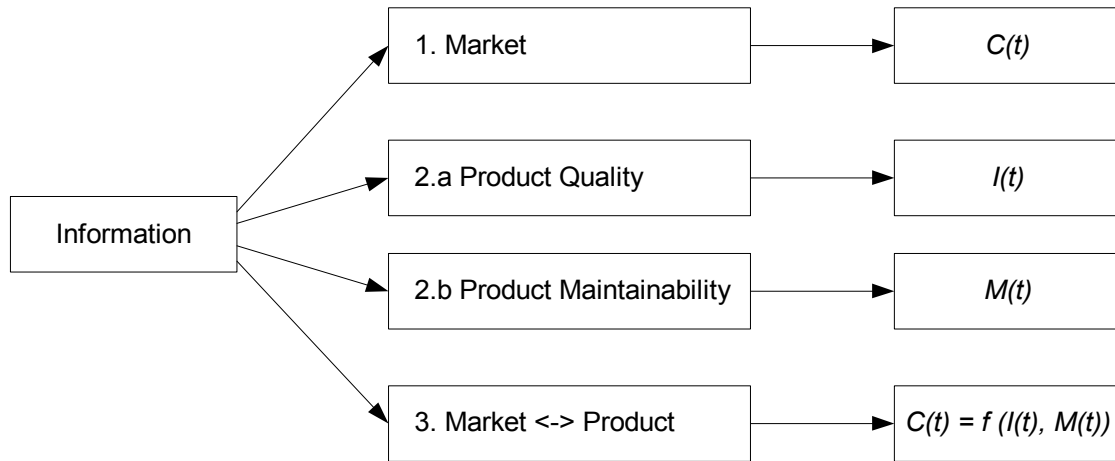


Figure 7-8: Main Sources of Information

Ideally, decision-makers would know the exact shapes of these functions and the relationships between the different functions. However, as discussed in Section 4.4.1, this is difficult, if not impossible. The case studies during the exploration phase reveal that software manufacturers face serious problems in estimating the exact shape of the development cost function $I(t)$, especially during the testing phase, and efforts to determine the shape of the post-release maintenance cost function $M(t)$ prior to a release decision were not found. To improve their decision-making process, the objective should be to search for sufficient information on the three sources of information, to enable decision-making without ‘too much’ uncertainty. Put differently, a decision-maker should look for the point of optimality. Below this point, uncertainty is high and might result in a software manufacturer missing a market window, or in unexpectedly high post-release maintenance costs. Beyond this point, the extra information leads to additional costs that outweigh the benefits [law of diminishing returns]. This point of optimality can probably not be determined precisely neither *ex ante* nor *post ante*. It is assumed the problems in determining this point, as raised by Gigerenzer (Section 7.2), are also valid for strategic software release decisions. So, instead of finding the point of optimality, software manufacturers will, in a practical setting, be forced to search for a zone of cost effectiveness: a bandwidth in which the marginal net asset value is equal or close to zero.

A final remark is reserved for changes to asset value, the development cost, and the maintenance cost functions. An increase of the asset value function, and decreases in both cost functions will lead to a positive change in the net asset value; increase in cash inflows and decrease in cash outflows. Improvement initiatives in the software industry focus primarily on decreasing the cost functions $I(t)$ and $M(t)$, with process improvement receiving much attention over the last decade (Section 1.3.2). In general, the common objective of these initiatives is to improve efficiency and effectiveness, with a decrease in development and maintenance cost, a reduction of the time-to-market, an increase in reliability and the maintainability level of the product, and an improvement in predictability accuracy. Collection, analysis and use of historical data from past projects are also promoted (see also Chapter 9). If successful, such improvements allow a software manufacturer to move the point of optimality, or zone of cost effectiveness, to the right: valuable information is obtained in less time and probably against less cost. This will enable them to make strategic software release decisions with less uncertainty. As argued in Section 7.2, a software manufacturer should be enabled to transform a

decision with complete uncertainty [point of optimality to the left] to a decision with informed uncertainty [point of optimality moving to the right] or, theoretically, even a decision with certainty [point of optimality completely to the right]. It is assumed that reducing the total uncertainty space will have two positive effects: the number of scenarios to be considered might be reduced, and the chance of occurrence of a scenario can possibly be better quantified with probability or possibility values. This makes the NPVI-method, as introduced in the previous Chapter, a better candidate for evaluating different release alternatives.

7.4 *Practices Identified*

In Section 5.3.1, the ‘*Release Information*’ process area is introduced reflecting the perspective of optimizing behaviour. This process area is concerned with the collection of information as input to the decision-making process. To reach the zone of effectiveness requires a pro-active attitude during product development. Up-front measures must be defined, and implemented, to have sufficient information available as input to the release decision-making process. In this study, the scope of information gathering is limited to information on the software product: both its quality and maintainability. Gathering market information and looking for relationships between external market-related and internal product-related information, should not be ignored, but is not taken into account.

Practices identified for collecting release information related to the software product are:

1. ***P-B1: Verification Definition.*** This practice concerns the identification of measures to determine the quality of the product. The exploratory case studies reveal that attention to the correct implementation of requirements is often postponed, for example, in the case of reliability, or even discarded, for example, in the case of maintainability (Section 4.4.2). From the case studies, it is concluded that the deployment and evaluation of non-functional requirements like reliability and maintainability is considered especially difficult (Section 4.4.2). The discussion on available theory (Sections 3.3.2 and 3.3.3) reveals limitations on the determination of reliability and maintainability. This leads to a lowered level of information perfection when the release decision is addressed. An improvement would be to explicitly evaluate such important non-functional requirements when selecting a design or architecture (see P-A4), when conducting reviews or inspections prior to testing. Also identified in the case studies, the use of standards might implicitly help to increase reliability and maintainability. Verification activities should be identified as early as possible, using a description of the *project deliverables* as input [from ‘*Release Definition*’ process area], preferably during, or directly after, the project launch, following the investment appraisal (see P-A1), as the effort required to implement them should be taken into account. If the project objectives change, the artefacts identified are subject to change.

The correct implementation of this practice is the responsibility of Development, assigned to execute the project. Maintenance & Exploitation may be involved to evaluate specific activities on maintenance aspects, e.g. reliability [short-term corrective maintenance] and maintainability [long-term adaptive/perfective maintenance].

2. ***P-B2: Verification Implementation.*** The definition of verification activities (see P-B1) alone is not sufficient, and the implementation of the defined activities should be ensured. As soon as verification activities have been identified (see P-B1), they can be scheduled for the most appropriate project phase to implement them. Project control should ensure satisfactory progress is made on the *implementation status* of the identified verification activities (see P-A2). This is an output from this process area.

The correct implementation of this practice is considered the responsibility of Development, being assigned to execute the project. Maintenance & Exploitation may be involved to evaluate whether implementations meet their requirements.

3. **P-B3: Artefact Identification.** This practice concerns the identification of measures to determine the maintainability of the product. It was found in the exploratory case studies (Section 4.4.2), that available documentation and its quality are unsatisfactorily low in many cases. This may lead to short-term problems [e.g. when user manuals are incomplete or do not exist at all] and/or long-term maintenance problems [e.g. when specifications or designs are missing or incomplete]. It is concluded that these artefacts should be identified as early as possible, using a description of the *project deliverables* as input, preferably during, or directly after, the project launch following the investment appraisal (see P-A1), as it requires effort to implement the artefacts. If the project objectives change, the artefacts identified are subject to change.

The correct implementation of this practice is the responsibility of Development, assigned to execute the project. Marketing may be involved to evaluate specific activities with regard to user documentation. Maintenance & Exploitation may be involved to evaluate specific activities on maintenance documentation.

4. **P-B4: Artefact Implementation.** The identification of artefacts (see P-B3) alone is not sufficient; the correct implementation and collection of the defined artefacts should be ensured. As soon as the artefacts are identified (see P-B3), the most appropriate project phase to implement the artefacts can be scheduled. Project control should ensure satisfactory progress is made on the *implementation status* of the artefacts (see P-A2). This is an output from the process area.

The correct implementation of this practice is the responsibility of Development, assigned to execute the project. Marketing and Maintenance & Exploitation may be involved to evaluate whether implementations meet their requirements.

In Figure 7-9, the data-flow diagram for the '*Release Information*' process area, combined with the previously discussed '*Release Definition*' process area is illustrated.

In Appendix F, a summary of this process area is given, including examples of supporting method(s) that can be used for the implementation of each practice.

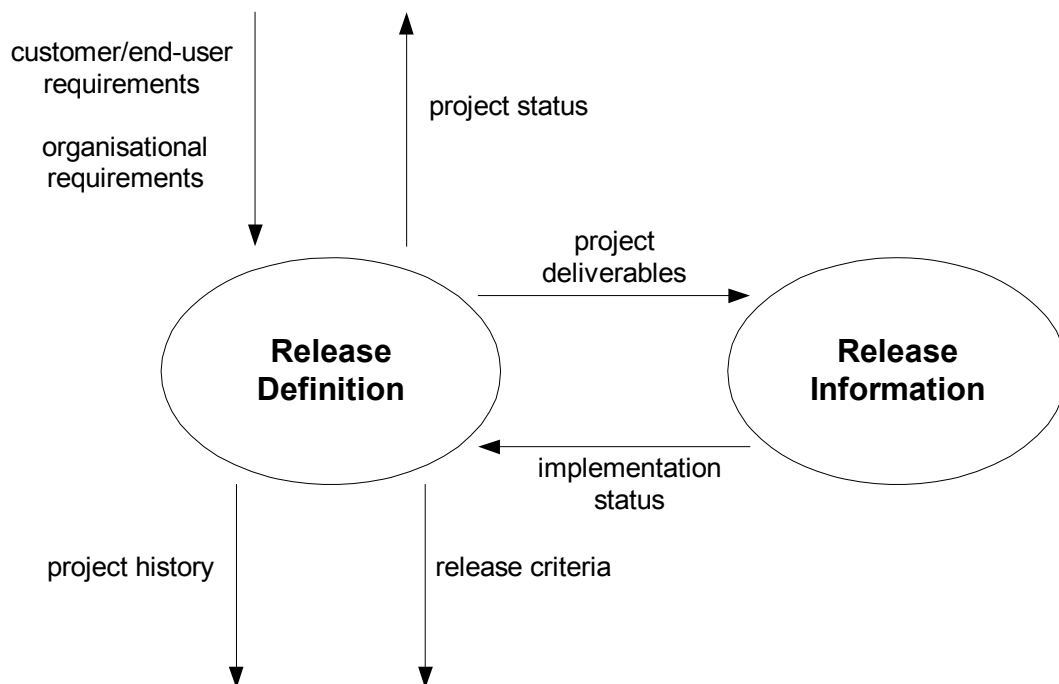


Figure 7-9: Extended Data-flow Diagram with '*Release Information*' Process Area

7.5 *Summary and Conclusions*

The ‘*Release Information*’ process area of the framework, as presented in Chapter 5, is discussed. The concept of maximizing behaviour is further investigated. Rather than assuming decision-makers possess all the relevant information for making choices, as in the previous Chapter, information itself is regarded as having a price in time and money. A decision-maker is now confronted with an even more complex problem. Apart from determining the demand and supply functions, a trade-off between the costs and benefits of searching for information should be made. This leads to optimizing behaviour instead of maximizing behaviour, but it was concluded that the difference is marginal. The primary objective of a decision-maker remains profit maximization, but it was also argued that finding the point of optimality, where the marginal value equals zero, is difficult, if not impossible.

The NPVI-method introduced in the previous Chapter is further refined for the formula used for calculating the net present value. This refinement enables the determination of different sources of information [market-related information, product-related information {product quality and product maintainability}] as input into the release decision-making process, and information on the relationship between external market-related and internal product-related information. The 2nd Secondary Research Question: ***‘To what extent can an optimal level of information be determined as input to the software release decision-making process?’*** remains unanswered for software release decisions. It is concluded that the problem of infinite regress, and the complexity of a practical context, prevents software manufacturers from precisely determining this point of optimality. Instead of deciding a point of optimality, software manufacturers are likely to be forced to search for a zone of cost effectiveness.

For the ‘*Release Information*’ process area, reflecting the perspective of optimizing behaviour, four practices are derived. The ‘P-B1 practice: Verification Definition’ concerns the definition of the activities to verify the quality of the product, namely the correct implementation of the stated functional and non-functional product requirements. When these activities are defined, the most appropriate project phase to implement them can be scheduled, under the ‘P-B2 practice: Verification Implementation’. ‘P-B3: Artefact Identification’ concerns the determination of the maintainability of the product: the identification of the supporting artefacts that need to be developed apart from the product. When these artefacts have been identified, the most appropriate project phase to implement them can be scheduled, as part of the ‘P-B4: Artefact Implementation’ practice.

In the next Chapter, limitations of optimizing behaviour [and thus maximizing behaviour] are discussed, taking into account the limitations of an individual decision-maker and the effects of group behaviour.